

SYSTEM AND METHOD FOR PROCESSING IMAGE DATA

Background

Image data is often stored and transmitted in a compressed format. The
5 compression of image data reduces the amount of memory utilized by storage of the data,
and also decreases the amount of time needed to transmit the image data across a
communications channel with a limited bandwidth compared to uncompressed data.

Many different types of data compression techniques are known. One type of
compression commonly used with image data is subsampling. Each pixel of color image
10 data in an image may be described as a set of three coordinates or components that locate
the color of the pixel in a selected color space. Subsampled image data is image data that
has at least one color component sampled at a lower spatial frequency than the other
color components.

One type of subsampling used in some color spaces is referred to as “4:2:2”
15 subsampling. This type of subsampling is typically used in color spaces that describe an
image in the terms of a luminance (or brightness) component and two chrominance (or
color) components. 4:2:2 subsampling involves sampling the chrominance components
at the same spatial frequency as the luminance component along the vertical direction,
but at one-half the spatial frequency of the luminance component in the horizontal
20 direction. Another type of subsampling is referred to as “4:2:0” subsampling. This is
similar to 4:2:2 subsampling, but involves sampling the chrominance components at one-

half the spatial frequency of the luminance component in both the vertical and horizontal directions.

In the decompression of subsampled image data, the subsampled data typically is read into a buffer configured to hold an entire frame of the data, and then the missing
5 chrominance values are calculated from the subsampled chrominance values. In some decompression systems, the data also may be read into a cache memory as it is being read into the buffer. Cache memory is a higher speed memory associated with many processors, and is used to increase processor speed. When the processor needs a value for a calculation, it checks the cache memory first. Where the value is not in the cache
10 memory, the processor must then find the value in the buffer stored in slower general memory. This is known as a “cache miss.” Significant numbers of cache misses may slow down processing speed.

Cache memory typically has a small size relative to general memory. For image data in a high-resolution format such as XGA, a buffer configured to hold an entire frame
15 of data may be larger than the cache. In this case, some image data in the cache memory may be written over by other image data as the data is read into the cache memory and the buffer. This may cause cache misses during the calculation of the missing chrominance values. The generation of too many cache misses may slow down image data processing, and thus may negatively affect the decompression and presentation
20 performance of subsampled video image data, which typically must be decompressed at a rate of approximately 25 frames/second or faster.

Summary

One embodiment provides a method of decompressing a set of subsampled image data. The method includes reading a first subset of the subsampled image data into a cache memory and into a buffer, wherein the buffer has an amount of memory equal to or
5 less than the cache memory, calculating chrominance values for at least some pixels of the subset of the subsampled image data to form decompressed image data, and outputting the decompressed image data.

Brief Description of the Drawings

Fig. 1 is a schematic depiction of an image data processing system according to an
10 embodiment of the present invention.

Fig. 2 is a flow diagram of a method of decompressing subsampled image data according to another embodiment of the present invention.

Fig. 3 is a schematic depiction of an exemplary decompression buffer containing 4:2:0 subsampled image data.

15 Fig. 4 is a schematic depiction of the buffer of Fig. 3, after calculating the missing chrominance values in the first line of the buffer.

Fig. 5 is a schematic depiction of the buffer of Fig. 3, after calculating some of the missing chrominance values in the second line of the buffer.

20 Fig. 6 is a schematic depiction of the buffer of Fig. 3, after calculating the remaining missing chrominance values in the second and third lines of the buffer.

Fig. 7 is a schematic depiction of the buffer of Fig. 3, after outputting the first two lines of decompressed image data from the buffer, resetting the first line pointer, and reading two new lines of compressed image data into the buffer.

Detailed Description of the Depicted Embodiments

5 Fig. 1 shows, generally at 10, a schematic depiction of an image processing system according to a first embodiment of the present invention. Image processing system 10 includes a display device 12 configured to display an image on a viewing surface 14. Display device 12 may be any suitable type of display device. Examples include, but are not limited to, liquid crystal display (LCD) and digital light processing (DLP) projectors
10 using either front or rear projection, television systems, computer monitors, etc.

Image processing system 10 also includes an image-rendering or processing device 16 associated with display device 12, and one or more image sources 18 in electrical communication (wired, radio, or any other suitable type of electrical communication) with image-rendering device 16. Image-rendering device 16 is
15 configured to receive image data transmitted by image sources 18, and to render the received image data for display by display device 12. Image-rendering device 16 may be integrated into display device 12, or may be provided as a separate component that is connectable to the display device. An example of a suitable image-rendering device is disclosed in U.S. Patent Application Serial No. 10/453,905, filed on June 2, 2003, which
20 is hereby incorporated by reference.

Image sources 18 may include any suitable device that is capable of providing image data to image-rendering device 16. Examples include, but are not limited to,

desktop computers and/or servers 18a, laptop computers 18b, personal digital assistants (PDAs) 18c, mobile telephones 18d, etc. Additionally, image sources 18 may communicate electrically with image-rendering device 16 in any suitable manner. In the depicted embodiment, each image source 18 communicates electrically with image-rendering device 16 over a wireless network 20. However, image sources 18 may also communicate with image-rendering device 16 over a wired network, over a wireless or wired direct connection, etc. or any mixture or combination thereof.

Image data may be compressed for transmission from image sources 18 to image-rendering device 16. As described above, one way in which the image data may be compressed is via subsampling, which is defined as the sampling of one or more color space components at a lower spatial frequency than the other color space components. Where image sources 18 are configured to provide chrominance-subsampled image data, such as 4:2:2 or 4:2:0 subsampled image data, to image-rendering device 16, the image-rendering device typically decompresses the subsampled image data by calculating missing chrominance values from the subsampled chrominance values.

Known decompression systems for decompressing subsampled image data typically read an entire compressed image into a decompression buffer before calculating the missing chrominance values. Often, the compressed image data is copied into a cache memory as it is read into the buffer. The cache memory and buffer on image-rendering device 16 are shown at 24 and 26, respectively, and a processor is shown at 22. However, because the size of a compressed image may be larger than the cache memory, some image data in the cache memory may be overwritten by other image data as the

compressed image data is copied into the buffer. The overwriting of image data in the cache memory may cause cache misses when the processor that is decompressing the image data looks for the overwritten data in the cache memory. The occurrence of too many cache memories may slow down image decompression to a detrimental extent.

5 To help avoid the occurrence of cache misses, image-rendering device 16 (or display device 12) may include a decompression buffer that is smaller than a cache memory located on the image-rendering device. Because cache memory is typically a relatively small memory, such a decompression buffer may also be smaller than a decompressed image. In other words, where the image data represents an image having
10 an $A \times B$ array of pixels (where A is a number of columns and B is a number of rows of pixels of an image), the decompression buffer may be configured to hold an $A \times C$ array of image data, wherein C is less than B . Such a buffer may be used to decompress a set of subsampled image data by reading the set of subsampled image data into the buffer and cache memory as a series of smaller subsets of image data. Each subset of image
15 data may be decompressed and output from the buffer before a new subset of the compressed image data is read into the decompression buffer. Because the decompression buffer is smaller than the cache memory, it is less likely that any image data in the cache memory will be overwritten while being used for decompression calculations.

20 The decompression buffer may have any suitable size. Generally, the smaller the decompression buffer is relative to the cache memory, the lower the likelihood of the occurrence of significant numbers of cache misses. Furthermore, the type of subsampled

image data to be decompressed in the decompression buffer and the types of calculations used to decompress the compressed image data may influence the size of the decompression buffer. For example, the missing chrominance components in 4:2:0 image data may be calculated differently depending upon whether the subsampled chrominance values are co-sited or non-co-sited. Co-sited chrominance values are positioned at the same physical location on an image as selected luminance values, while non-co-sited chrominance values are positioned interstitially between several associated luminance values. The missing chrominance values of 4:2:0 co-sited image data may be calculated from subsampled chrominance values either on the same line as the missing values, or on adjacent lines, depending upon the physical location of the missing chrominance value being calculated. Thus, a decompression buffer for decompressing 4:2:0 image data, which has lines of data having no chrominance values, may be configured to hold more than one line of image data to allow missing chrominance values to be calculated from vertically adjacent chrominance values.

Fig. 2 shows, generally at 100, an embodiment of an exemplary method for decompressing 4:2:0 co-sited image data. Method 100 utilizes a three-line decompression buffer (i.e. $C = 3$ for the above-described $A \times C$ decompression buffer) that allows missing chrominance data to be calculated from both horizontally and vertically adjacent subsampled chrominance values. Method 100 is typically embodied in code stored in memory 24 on image-rendering device 16 and is executed by processor 22 on image-rendering device 16, but also may be embodied in code that is stored in memory projector device 12, or elsewhere on network 20.

Method 100 first includes receiving a set of subsampled image data at 102. Because a single set of image data is often compressed in multiple ways for transmission, the set of subsampled image data received at 102 may be received from another decompressor program. Alternatively, the set of subsampled image data received at 102
5 may be received directly from one of image sources 18. Next, an initial subset of the subsampled image data is read into the three-line decompression buffer at 104. The initial subset of the subsampled image data read into the decompression buffer at 104 typically is as large as the decompression buffer. In other words, where the decompression buffer is configured to hold three lines of image data, a three-line subset
10 of the subsampled image data is read into the decompression buffer at 104. The initial subset of image data may be taken from any desired portion of the set of subsampled image data. For example, either the top three lines, the bottom three lines, or any desired intermediate three lines of the A x B image data may be read into the decompression buffer at 104 for decompression.

15 After reading the initial subset of subsampled image data into the three-line buffer at 104, the missing chrominance values are calculated from the subsampled chrominance values. The missing chrominance values may be calculated from the subsampled chrominance values in any suitable manner, and in any suitable order. In the exemplary embodiment of Fig. 2, the missing chrominance values on the first line of the buffer are
20 first calculated at 106, and those on the third line of the buffer are calculated next at 108. These calculations are illustrated schematically in Fig. 3, which depicts the three-line

decompression buffer at 200, the first line of the decompression buffer at 202, the second line of the buffer at 204, and the third line of the buffer at 206.

Each pixel of the image is represented by a single location 208 in decompression buffer 200. Each location 208 typically has enough memory to hold a luminance value, two chrominance values, and any other data associated with a single pixel of decompressed image data. Luminance values are represented by squares 210 at each location 208 in the buffer containing a luminance value (i.e. all locations for 4:2:0 subsampled image data), and the co-sited chrominance values are represented by circles 212 at the memory locations in the buffer containing the luminance values with which the subsampled chrominance values are co-sited. It can be seen that the chrominance values of the data in decompression buffer 200 are sampled at only half the frequency of the luminance values in both vertical and horizontal directions. A software pointer, represented schematically by arrow 214 in Fig. 3, may be provided to indicate to the decompressor program which line of buffer 200 is considered the first line for purposes of decompression calculations. Pointers (not shown) may also be used to indicate which lines of buffer 200 are the second and third lines.

The calculation of the missing chrominance values in first line 202 and third line 206 from the subsampled chrominance values in the first line is indicated schematically by arrows 216. In the depicted embodiment, the missing chrominance values in first line 202 are calculated from immediately horizontally adjacent subsampled chrominance values, but may also be calculated from vertically adjacent chrominance values where suitable. While each missing chrominance value is depicted as being calculated from the

subsampled chrominance value to the immediate left, it will be appreciated that each missing chrominance value also may be calculated from the subsampled chrominance value to the immediate right, as depicted by arrows 216'. Where the missing chrominance values are calculated from subsampled chrominance values to the immediate right, the final chrominance value in a line of data may be filled in by copying the second-to-last value in the line. The actual calculations used to determine the missing chrominance values are well known to one skilled in the art, and are thus not described in further detail.

While Figs. 2 and 3 depict the missing chrominance values on the third line being calculated before the missing chrominance values on the second line, it will be appreciated that the missing chrominance values on the second line may be calculated either before or after the missing chrominance values on the third line.

The missing chrominance values on second line 204 and third line 206 of buffer 200 may be calculated in any suitable manner. Fig. 4 depicts a schematic representation of an exemplary calculation of a first portion of the missing chrominance values in second line 204 of decompression buffer 200 from the subsampled chrominance values in third line 206. Because the data in buffer 200 is 4:2:0 subsampled data, every other line of the data has no chrominance values. Thus, the missing chrominance values in these lines may be calculated from subsampled chrominance values in the lines immediately above and/or below. In Fig. 4, alternating missing chrominance values 220 in second line 204 of decompression buffer 200 are calculated from chrominance values 222 located immediately below in third line 206 of the decompression buffer, as depicted by arrows

223. Alternatively, the missing chrominance values are calculated from the chrominance values located immediately above the missing chrominance values, as indicated by dashed-line arrows 223'. Where the missing chrominance values in second line 204 are calculated from the subsampled chrominance values in the line immediately below, and wherein the last line in a set of image data includes missing chrominance values, the missing chrominance values may be calculated or copied from the line of data immediately above. As with the missing chrominance values on first line 202 and third line 206, the actual calculations used to calculate the missing chrominance values in second line 204 are known to one skilled in the art, and thus are not described in detail herein.

Fig. 5 shows a schematic depiction of the calculation of the remaining missing chrominance values on second line 204. The remaining missing chrominance values on second line 204 are indicated at 224, and are calculated from the chrominance values previously calculated for positions 220. As with the calculations of the other missing chrominance values described above, the actual calculations used to fill in the remaining missing chrominance values on second line 204 are known to one skilled in the art, and are thus not discussed in detail.

Due to the relatively small size of decompression buffer 200 compared to the amount of data in an entire image, it is likely that few cache misses will be generated by the decompression calculations described above for the initial subset of image data, even where the image encoded by the data is in a high resolution format. Furthermore, because decompression buffer 200 holds three lines of image data, subsampled

chrominance values may be used to calculate missing chrominance values in vertically adjacent positions. It will be appreciated that the three-line configuration of decompression buffer 200 is merely exemplary, and that a decompression buffer according to the present invention may be configured to hold any other suitable number
5 of lines of image data.

Referring again to Fig. 2, after completing the calculation of the missing chrominance values in the initial subset of subsampled image data, the decompressed image data is output from the buffer at 112. The decompressed image data is typically output to an output buffer configured to hold an entire decompressed image, and may be
10 output in any suitable manner. For example, all three lines of the image data may be output, or a smaller subset of image data may be output. In the embodiment of Fig. 2, only the decompressed data from the first and second lines are output. It will be appreciated that other steps may be performed after decompressing the subset of subsampled image data before outputting the decompressed data. For example, a color
15 conversion from a luminance/chrominance color space to an RGB color space may also be performed before outputting the data.

Outputting two lines of data, instead of all three lines of data, from decompression buffer 200 may offer advantages over outputting all three lines of data. For example, outputting two lines of data and resetting the third line to be the first line always results in
20 the subsampled chrominance values occupying the first and third lines of the buffer (assuming lines of image data are read into the decompression buffer in an incremental

order). This may help to simplify the decompression calculations where a plurality of subsets of the subsampled image data are decompressed.

At this point, the decompressor program determines at 114 whether the set of subsampled image data is completely decompressed. Where the set of subsampled image data is not completely decompressed, first line pointer 214 is reset at 116 to point at third line 206, as shown schematically in Fig. 6. This resets third line 206 as a first line 202' of buffer 200 for performing decompression calculations. At the time first line pointer 214 is reset, first line 202 is also reset as second line 204' of the buffer, and second line 204 is reset as third line 206' of the buffer. This is a circular ring buffer of three lines.

Resetting first line pointer 214 allows the third line to be reset as the first line without shifting the positions of any of the third line image data values in memory. Thus, resetting first line pointer 214 may be a faster and more efficient method of resetting the third line as the first line than shifting image data values within the decompression buffer. Alternatively, the third line of the buffer may be reset as the first line of the buffer by shifting the image data values in the third line to the first line.

After resetting at 116 the assignments of the lines in the buffer to first, second and third lines 202', 204' and 206' respectively, a new two-line subset of subsampled image data is read at 118 into second line 204' and third line 206', and method 100 loops back to step 108 to calculate the missing chrominance values on the third line at 108 and on the second line at 110. These calculations are indicated schematically by arrows 230 in Fig. 7. Method 100 continues to loop through steps 108-118 until the set of subsampled image data has been completely decompressed. As this loop is repeated, pointer 214

rotates repeatedly through the lines of the buffer, moving from first line 202, to third line 206, and back to first line 202.

Once the set of subsampled image data has been completely decompressed, the loop is broken at step 114, and the decompressed data in the third line of decompression buffer 200 is output from the buffer at 120. It will be appreciated that, depending upon the format of the image encoded in the set of subsampled image data, the final subset of image data read into buffer 200 may have only one line of data, rather than two. In this case, the missing chrominance values of that line may be calculated from the decompressed data in the current first line of the buffer, and then both lines may be output.

While decompression buffer 200 is disclosed as being configured to hold an $A \times C$ array of image data, wherein $C = 3$, it will be appreciated that other buffer configurations may also be used. For example, where the image data is arranged in an $A \times B$ array, a decompression buffer according to the present invention may have a $C \times B$ configuration, wherein C is less than A , or even a $C \times D$ configuration, wherein C and D are less than A and B , respectively.

Furthermore, although the present disclosure includes specific embodiments, specific embodiments are not to be considered in a limiting sense, because numerous variations are possible. The subject matter of the present disclosure includes all novel and nonobvious combinations and subcombinations of the various elements, features, functions, and/or properties disclosed herein. The following claims particularly point out certain combinations and subcombinations regarded as novel and nonobvious. These

claims may refer to “an” element or “a first” element or the equivalent thereof. Such claims should be understood to include incorporation of one or more such elements, neither requiring nor excluding two or more such elements. Other combinations and subcombinations of features, functions, elements, and/or properties may be claimed
5 through amendment of the present claims or through presentation of new claims in this or a related application. Such claims, whether broader, narrower, equal, or different in scope to the original claims, also are regarded as included within the subject matter of the present disclosure.